



KANDIDAATINTYÖ

Sulautettujen suorittimien muistityypit ja niiden käyttö

Eetu Parkkinen

Ohjaaja: Timo Rahkonen

**ELEKTRONIIKAN JA TIETOLIIKENNETEKNIIKAN
TUTKINTO-OHJELMA
2020**

Eetu Parkkinen (2020) Sulautettujen suorittimien muistityypit ja niiden käyttö.
Oulun yliopisto, Elektroniikan ja tietoliikennetekniikan tutkinto-ohjelma.
Kandidaatintyö, 22 s.

TIIVISTELMÄ

Tässä työssä käydään aluksi läpi sulautetuissa prosessorilaitteissa käytettäviä muistityyppejä ja niiden toteutustapoja. Muisteja on haihtuvia ja haihtumattomia. Nopeus ja hinta erottavat haihtuvat muistit toisistaan. Haihtumattomien muistien erot tulevat lähinnä ohjelmointiajasta ja ohjelmointikertojen lukumäärästä, mutta pieniä eroja on myös virrankulutuksessa. Työn lopussa käydään esimerkinomaisesti läpi PIC24F-tyyppisen mikrokontrollerin muistirakenne, joka koostuu flash-ohjelmamuistista ja EEPROM-datamuistista, ja kerrotaan muutama sana reaaliaikakellopiireistä ja niiden käyttämästä muistista. Reaaliaikakellopiireissä on monesti muutamia kymmeniä tavua sekä haihtumatonta, että haihtuvaa muistia.

Avainsanat: haihtuva muisti, haihtumaton muisti, SRAM, DRAM, FRAM, flash, EEPROM, RTC.

Eetu Parkkinen (2020) Memory types of integrated processors and their usage.
University of Oulu, Degree Programme in Electronics and Communications
Engineering, Bachelor's Thesis, 22 p.

ABSTRACT

This work first reviews the types of memory used in integrated processor devices and implementations of the memory types. Memories are volatile or non-volatile. Speed and price separate volatile memories from each other. The differences in non-volatile memories come mainly from the programming time and the number of programming times. There are also small differences in power consumption. At the end of the work, the memory structure of the PIC24F type microcontroller is reviewed as an example. This memory structure consists of flash program memory and EEPROM data memory. Also, the study has few words about real-time clock circuits and the memory used by them. Real-time clock circuits often have a few dozen bytes of both non-volatile and volatile memory.

Keywords: volatile memory, non-volatile memory, SRAM, DRAM, FRAM, flash, EEPROM, RTC.

SISÄLLYS

TIIVISTELMÄ.....	2
ABSTRACT	3
SISÄLLYS.....	4
1. JOHDANTO	5
2. SULAUTETUISSA JÄRJESTELMISSÄ OLEVA DATA	6
3. HAIHTUVIA MUISTITYYPPEJÄ.....	7
3.1 SRAM.....	7
3.2 DRAM.....	8
4. HAIHTUMATTOMIA MUISTITYYPPEJÄ.....	9
4.1 FLASH.....	9
4.2 EEPROM.....	10
4.3 FRAM JA SEN KÄYTTÖ MSP430-MIKROKONTROLLERISSA....	11
5. PIC24F:N MUISTIRAKENNE	14
5.1 MIKROKONTROLLERIN YLEISKATSAUS.....	14
5.2 MUISTIRAKENNE.....	14
5.3 FLASH-OHJELMAMUISTI.....	16
5.4 EEPROM-DATAMUISTI	17
6. REAALIAIKAKELLOPIIRI JA SEN MUISTI.....	18
7. POHDINTA	19
8. YHTEENVETO	20
9. LÄHTEET	21

1. JOHDANTO

Mikrokontrollereissa on prosessori, *Central Prosessing Unit* (CPU), kello, ohjelmamuisti, muunneltavaa muistia ja erilaisia sisään- ja ulostuloja. Useimmat mikrokontrollerit eivät tarvitse ulkoisia muistielementtejä, sillä ne ovat sulautettuna piiriin muiden komponenttien kanssa [1]. Osa mikrokontrollereista tarvitsee vain RAM-muistia, *Random Access Memory*, tai ROM-muistia, *Read Only Memory*, toimiakseen, mutta usein käytössä on myös haihtumatonta muistia.

Tässä työssä käsitellään aluksi haihtuvia RAM-muisteja: SRAM-muistia ja DRAM-muistia ja erityisesti haihtumatonta FRAM-muistia ja sen käyttöä Texas Instrumentsin valmistamassa MSP30-mikrokontrollerissa. Muita haihtumattomia muistityyppejä, jotka esitellään tässä työssä ovat flash-muisti ja ROM-muisteista erityisesti EEPROM-muisti. Muistityyppien esittelyjen jälkeen avataan niiden käyttöä Microchip Technologyn PIC24F-mikrokontrollerissa. Lopuksi käydään läpi reaaliaikakellopiirin toimintaa ja sen muistinkäyttöä.

Tämän työn tarkoitus on selvittää miten eri muistityypit rakentuvat ja mitä eroja niillä on. Lisäksi selvitetään mihin tarkoituksiin eri muistit soveltuvat parhaiten ominaisuuksiensa takia. Tämä työ on kirjallisuustutkimus, ja materiaalia on kerätty sekä verkossa julkaistuista tutkimuksista ja kirjallisuudesta, että eri mikrokontrollerivalmistajien sivuilta ja valmistettujen laitteiden datalehdistä.

Olen valinnut tämän aiheen, koska muistit ovat tärkeä osa elektroniikkaa ja niissä tapahtuvaa datankäsittelyä. Lisäksi muistit kehittyvät edelleen ja niiltä vaadittavat ominaisuudet muuttuvat jatkuvasti. Etenkin IT-sovellukset pakottavat tulevaisuuden muistit omaamaan entistäkin nopeamman pääsyn dataan, jotta tieto liikkuu nopeasti, ja suuremman tiheyden, jotta voidaan tallentaa yhä suurempia määriä dataa yhteen muistiin. Tämän kaiken lisäksi laitteella tulee olla pieni virrankulutus. [12]

2. SULAUTETUISSA JÄRJESTELMISSÄ KÄSITELTÄVÄ DATA

Sulautetuissa järjestelmissä käsitellään monenlaista dataa. Osa datasta on järjestelmässä sellaista, jota voi vain lukea. Tällaista dataa ovat esimerkiksi alkulatauskoodi ja ohjelmakoodi, jota ei haluta päivittää. Data siis sisältää parametreja, joita käyttäjä ei pääse muokkaamaan. Datalle, jota ei voi muokata, on ominaista tarve säilyttää sisältönsä, vaikka järjestelmästä katkaistaisiinkin virta. Datan tallentamiseen käytettävillä muisteilla haihtumattomuus on tärkeä ominaisuus.

Datan halutaan yleensä olevan haihtumatonta, mutta haihtuvallekin datalle on paikkansa. Haihtuva muisti on nopeaa, joten sinne halutaan tallentaa tieto, jota tarvitaan suurimmalla todennäköisyydellä. Tällaista haihtuvaa ja erittäin nopeasti liikuteltavaa dataa on tallennettuna lähelle suoritinta välimuistiin. Ohjelman toiminnassa dataa tallennetaan väliaikaisesti haihtuvana, jos sitä tullaan vielä käsittelemään ohjelman toiminnan aikana.

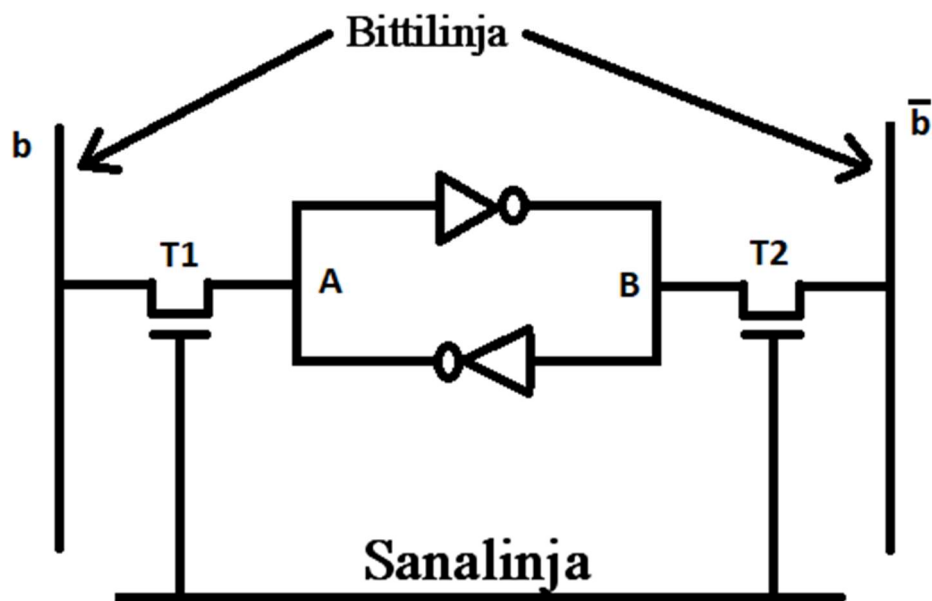
Muita datatyyppejä, joita voi muokata ja uudelleenkirjoittaa, ovat kalibrointiarvot sekä päivitettävä ohjelmakoodi. Se minkälaisiin muisteihin eri datatyypit tallennetaan, riippuu pitkälti siitä, kuinka usein ja nopeasti kyseessä olevaan dataan halutaan päästä käsiksi ja kuinka pitkään sen pitää olla tallennettuna. Ihannetapauksessa datan tallentamiseen käytettäisiin aina suurta määrää hyvin nopeaa muistia, mutta käytännössä tämä ei ole mahdollista: suurikapasiteettisesta muistista on vaikea tehdä nopeaa, ja nopea muisti on kallista. Tästä syystä muistista tehdään hierarkia, jossa mahdollisimman lähellä suoritinta on nopein, mutta pienin muisti, kauempana vähän suurempi ja hitaampi muisti, ja kaikkein kauimmaksi on sijoitettuna suurin ja hitain muisti, massamuisti, johon tallennetaan pysyvästi tallennettu data.

3. HAIHTUVIA MUISTITYYPPEJÄ

3.1 SRAM

Yleisesti RAM on lyhytkestoinen muisti, jota käytetään käynnissä olevien ohjelmien datan säilyttämiseen. Eri RAM-muistityypit ovat yleensä haihtuvia, eli niiden sisältö tyhjenee, kun virta katkaistaan. Joissain mikrokontrollerissa voi olla vain RAM-muistia, kun on tarve ladata ohjelma muistiin ja ohjata prosessori aloittamaan tämän ohjelman suorittaminen jossain ohjelman tulopisteessä.

SRAM-muisti eli *Static Random Access Memory* on suhteellisen kallis, mutta nopea muisti, joka käyttää pieniä kiikkuja bittien varastointiin. SRAM-muistialkio voidaan rakentaa esimerkiksi kuvan 1 osoittamalla tavalla [2]. Siinä kahdella NOT-portilla muodostettu kiikku on yhdistetty transistoreilla T1 ja T2 kahteen bittilinjaan. Nämä kaksi transistoria toimivat kytkiminä, jotka voidaan avata tai sulkea osoitetekooderin ohjaaman sanalinjan kautta. Esimerkiksi muistialkio on tilassa 1, kun pisteen A looginen arvo on 1 ja pisteen B looginen arvo on 0. Tila muuttuu vasta, kun sanalinja aktivoidaan. SRAM ei tarvitse DRAM:in tavoin virkistämistä sisällön säilyttämiseen niin kauan, kun virtaa syötetään. SRAM:ia on lähes kaikissa mikrokontrollereissa [3].



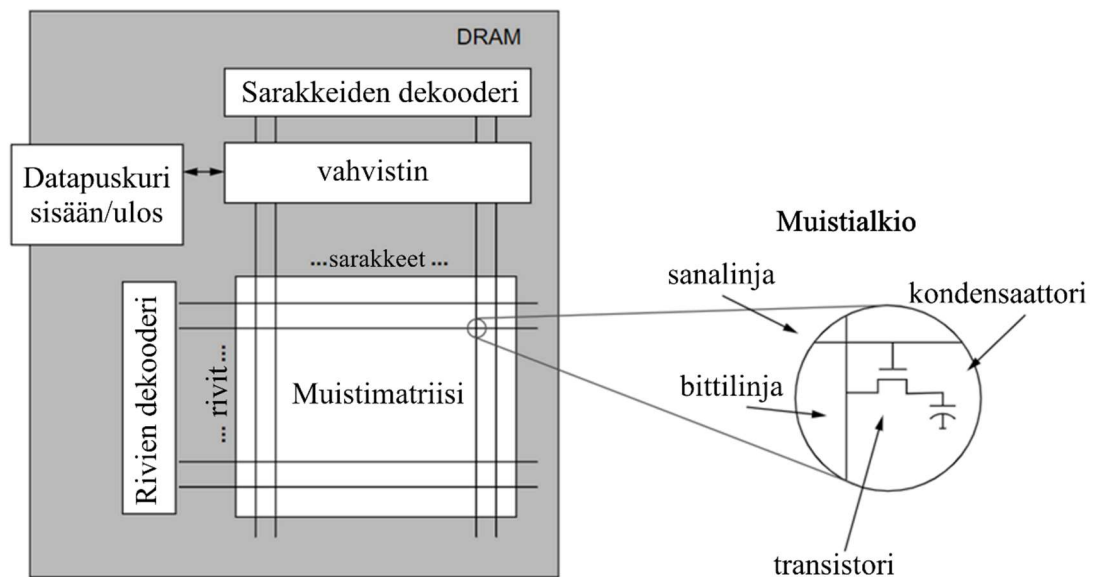
Kuva 1. SRAM-muistialkion rakenne.

SRAM-muistia käytetään suuren luku- ja kirjoitusnopeutensa vuoksi järjestelmissä välimuistina, jonka varastoimaa dataa prosessori käyttää säännöllisimmin. Erityisesti isot muuttujat sijoitetaan SRAM-muistiin. Välimuisti on järjestelmän nopein muisti, sillä sen täytyy pysyä prosessorin mukana, kun se hakee ja suorittaa käskyjä. Nopein osa välimuistista sijaitsee prosessorin sisällä olevassa muistialueessa, joka sisältää useita rekistereitä. Nämä muistihierarkiassa huipulla olevat rekisterit mahdollistavat nopean pääsyn dataan.

Hitaampia muotoja välimuistista ovat L1- ja L2-tason välimuistit. Näistä kahdesta nopeampi, mutta vain kymmenien kilotavujen kokoinen L1-välimuisti sijaitsee suorittimessa itsessään. Satojen ja jopa tuhansien kilotavujen kokoinen L2-välimuisti liitettyinä suorittimeen, mutta se sijaitsee suorittimen ulkopuolella. Muitakin välimuistitasoja käytetään, mutta harvemmin. Nämä välimuistityypit sijaitsevat kuitenkin yleensä eri piirillä kuin itse prosessori, ja niissä saatetaan käyttää jotain DRAM-pohjaista muistityyppiä.

3.2 DRAM

DRAM-muisti eli *Dynamic Random Access Memory* on SRAM-muistia hitaampi, mutta halvempi muistityyppi. Hintaero tulee muistissa tarvittavien transistoreiden määrästä, joka on DRAM:lla pienempi. DRAM käyttää bittien varastointiin yksittäisiä transistori-kondensaattoripareja, jotka ajan kuluessa menettävät varauksensa. Kuvassa 2 näkyy, kuinka muistirakenteessa transistori-kondensaattoriparit on järjestetty riveihin ja sarakkeisiin muodostaen bitti- ja sanalinjat. Informaation tallentamiseksi alkioon, transistori kytketään päälle ja sopiva jännite asetetaan bittilinjalle. Tällöin kondensaattoriin varastoituu tietyn suuruinen varaus.



Kuva 2. DRAM-muistin rakenne.

DRAM:in dynaamisuus tulee siitä, että muistia pitää jatkuvasti virkistää sisällön säilyttämiseksi. DRAM:in tyypillinen virkistysaika on muutamasta millisekunnista satoihin millisekunteihin [4]. Virkistämisessä muistista luetaan haluttu bitti, joka kirjoitetaan uudelleen takaisin muistiin. DRAM:ia ei käytetä sulautetuissa järjestelmissä, ja sen käyttökohde on enimmäkseen keskusmuistina PC:ssä.

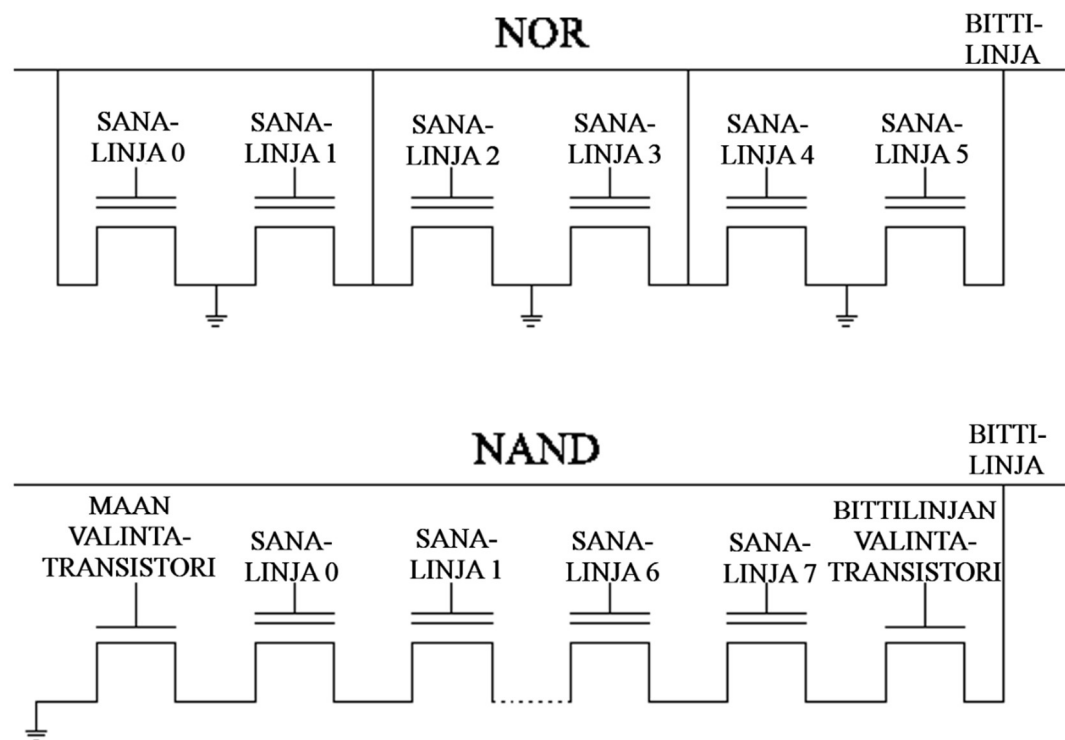
4. HAIHTUMATTOMIA MUISTITYYPPEJÄ

4.1 FLASH

Flash-muisti on puolijohdemuisti, joka on käytössä joustava ja nopea verrattuna ROM-muistiin [5]. Flash-muistia on kahta päätyyppiä NAND- ja NOR-flash-muistia. Yksittäiset flash-muistialkiot muodostuvat kelluvahilaisista MOSFET:sta, jotka varastoivat informaation.

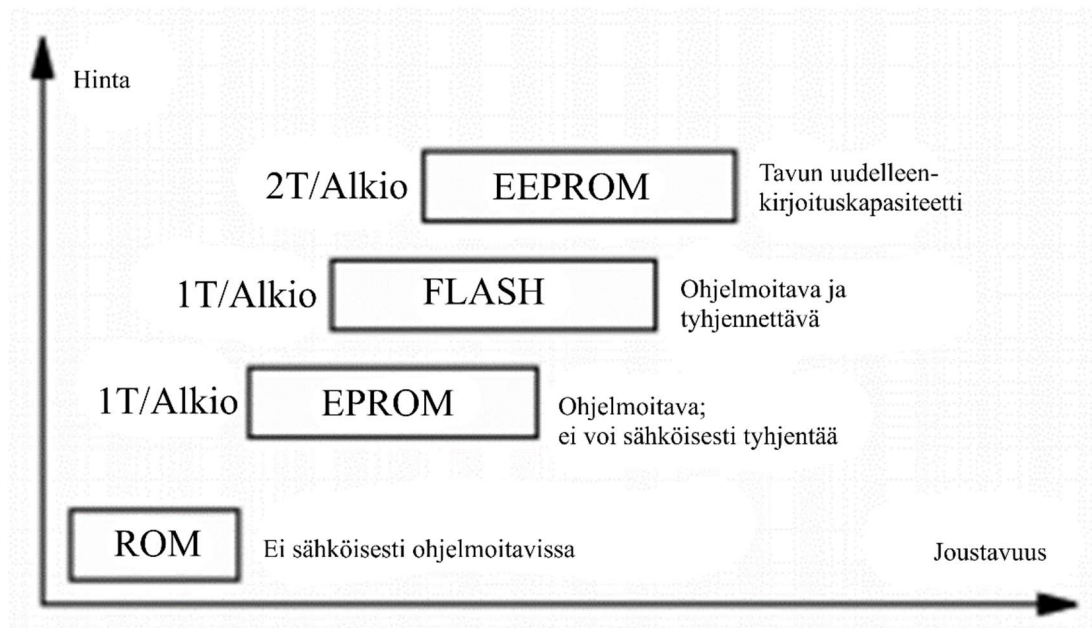
NOR-flash-muistissa jokainen muistialkio on kytketty toisesta päästä suoraan maahan ja toisesta päästä bittilinjalle. Tällaisen flashin NOR-nimitys tulee siitä, että yhdenkin sanalinjan varauksen ollessa ylhäällä, eli transistorin VT-arvon yläpuolella, bittilinjan ulostulon varaus on päinvastoin alhaalla. Tällöin NOR-flash-muistialkion oletustila on looginen ”1”.

NAND-flash-muistin transistorit on kytketty siten, että ne muodostavat NAND-portin: muistin transistorit ovat sarjassa, ja bittilinjan jännite on alhaalla vain, kun kaikkien sanalinjojen varaus on transistorin VT-arvoa suurempi. Alla olevassa Kuva 3. NOR- ja NAND-flash-muistien rakenteet.



Kuva 3. NOR- ja NAND-flash-muistien rakenteet

Haihtumattomia muisteja voidaan vertailla joustavuuden ja hinnan mukaan (Kuva 4). Joustavuudella tarkoitetaan, onko ohjelmointimahdollisuutta ja kuinka monesti ja helposti muisti voidaan tyhjentää ja uudelleenohjelmoida. Hintaan taas vaikuttaa muistin monimutkaisuus ja muistialkion koko. Flash-muistin etuina ovat nopeuden lisäksi haihtumattomuus ja pieni koko. Lisäksi muisti voidaan sähköisesti tyhjentää ja uudelleenohjelmoida.



Kuva 4. Eri haihtumattomien muistien laatuvertailua joustavuus-hinta -koordinaatistossa.

Flash-muisti on erityisen hyvä hintalaatu-suhteeltaan pienissä määrissä [6]. Tyypillisesti flash-muistia käytetään alkulatauskoodin ja parametrien säilyttämisen lisäksi ohjelmistomuistina [7].

4.2 EEPROM

On mahdollista, että on vain ROM-muistilla toimiva mikrokontrolleri. Tällöin ei voi käyttää ohjelmia, jotka käyttävät aliohjelmien kutsuja tai keskeytyksiä, sillä ROM-muistiin ei voi kirjoittaa. Lisäksi kaikkien muistissa olevien muuttujien on sovittava suorittimen rekistereihin. *Electrically Erasable Programmable Read Only Memory*, EEPROM, on ROM-muistityyppi, jota voidaan sähköisesti tyhjentää ja uudelleenohjelmoida. EEPROM-muisti rakentuu siihen pohjautuvan flash-muistin tavoin kelluvahilaisten MOSFET:ien riveistä.

EEPROM-yksiköiden ohjelmointi ja piiristä poistaminen tapahtuvat käyttämällä erityisiä ohjelmointikäskyjä. EEPROM-laitteet käyttävät sarja- tai rinnakkaisliitäntää datan sisään- tai ulostuloissa. EEPROM-muistin tyypillinen protokolla muodostuu kolmesta vaiheesta: OP-koodi-, osoite- ja datavaiheista. OP-koodi määrittää käskyt, joita osoite- ja datavaiheet noudattavat toiminnassaan.[8]

EEPROM-muisti tyhjennetään yhden bitin sijaan valittu alue kuten tavu tai tavurivi kerrallaan ilman vaikuttamista koko muistirakenteeseen. Lisäksi muistiin voidaan liittää termi "FLASH", mikä tarkoittaa muistin tyhjentämistä isommissa lohkoissa. Tällaista flash-EEPROM-muistia ei ole mahdollista tyhjentää yksittäinen tavu kerrallaan joten, kun muistin tyhjentäminen tapahtuu, kaikki EEPROM-muistialkiot

flash-rivillä tyhjentyvät. Tällainen rakenne vähentää tarvittavien transistoreiden määrää verrattuna tavuittain tyhjennettävään EEPROM-muistiin. [1]

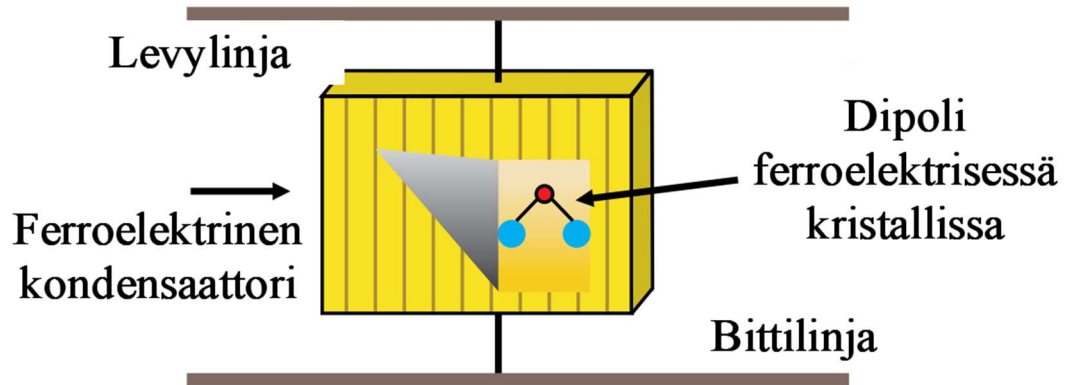
EEPROM vaatii muihin edellä esitettyihin muisteihin verrattuna suhteellisen paljon pinta-alaa muistialkiota kohden ja sillä on monimutkaisempi rakenne muihin nähden, kuten kuva 4 havainnollistaa. Toisaalta muisti on hyvin joustava, sillä se uudelleenohjelmoitavuuden lisäksi säilyttää sisältönsä, vaikka virransyöttö mikrokontrolleriin lopetettaisiin. Muistin yhtenä sovelluksena on käyttää sitä kalibroituvakioiden säilyttämiseen laitteissa, jotka kuluvat käytössä. Käytössä mikrokontrollerin CPU asettaa bitit ohjausrekisterissä laittamaan päälle ohjelmointijännitteen, mikä mahdollistaa EEPROM-muistin ohjelmoimisen. Tämän jälkeen CPU kirjoittaa ohjelmoitavalle EEPROM-muistialkiolle. [1] Monissa mikrokontrollereissa käytetään EEPROM-muistia parametrien säilyttämiseen flash-muistin toimiessa sen rinnalla ohjelmistomuistina.

4.3 FRAM JA SEN KÄYTTÖ MSP430-MIKROKONTROLLERISSA

Yleensä yksi muistityyppi ei kykene yksinään täyttämään laitteen koko tallennustilalta vaadittavia ominaisuuksia. Tästä syystä sovelluksissa käytetään yksittäisten muistityyppien yhdistelmiä. Esimerkiksi matkapuhelimeissa pienitehoista SRAM-muistia ja sähköisesti ohjelmoitavaa EEPROM-lukumuistia käytetään yhdessä tallennetun datan käsittelemiseen. Tästä on muistien yhdistelemisestä on useita haittoja: suuri tilavuus ja monimutkainen ajoituksen säätö. Muistiominaisuuksiensa takia ferrosähköisissä muisteissa on eniten potentiaalia olemaan haihtumattomien muistityyppien joukosta se, joka kykenee tyydyttämään yksinään kaikki yhden laitteen asettamat muistitarpeet.

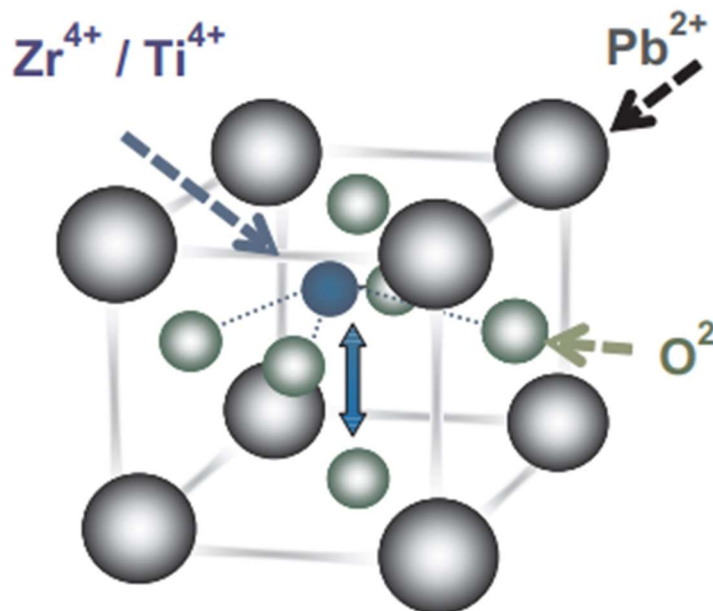
FRAM, *Ferroelectric Random Access Memory*, on haihtumaton RAM-muisti. FRAM-muistiteknologia yhdistää flash-muistin haihtumattomuuden ja DRAM-muistin luku- ja kirjoitusnopeuden käyttäessään silti hyvin vähän energiaa verrattuna muihin muistityyppeihin. Muistin haihtumattomuus saavutetaan käyttämällä ferrosähköistä kidettä, jonka varastointitila perustuu vapaiden atomien tiloihin kiteellä.

FRAM Muistialkio



Kuva 5. FRAM muistialkio.

FRAM-muistialkio muodostuu bitti- ja levylinjojen väliin liitetystä kondensaattorista, joka koostuu ferrosähköisestä kiteestä (Kuva 5) [9]. Kide on yleensä lyijy-zirkonium-titanaattia, PZT, ja sen rakenne ja toiminnallisuus ovat havainnollistettuina Kuva 6 [10]. Kondensaattorin dipolin orientaatio määrittää kumpi "1" vai "0" on varastoituna. Dipolin orientaatio voidaan määrätä asettamalla jännitteen kiteen yli, jolloin sähkökenttä saa kiteessä olevan vapaan zirkoniumatomin liikkumaan yhdestä stabiilista tilasta toiseen. Tämä stabiili tila pysyy, vaikka kiteen yli oleva jännite poistettaisiin, mikä tekee FRAM-muistista haihtumattoman.



Kuva 6. PZT-kiteen rakenne.

Muihin edellä esiteltyihin muistityyppeihin verrattuna FRAM-muistilla on useita etuja. Kuten DRAM-muistilla, FRAM:lla on lyhyt kirjoitusaika, joka on tuhannesosa EEPROM- tai flash-muistien kirjoitusaikoihin verrattuna. Lisäksi tallennus tarvitsee matalan 1,5 V jännitteen, kun esimerkiksi EEPROM tarvitsee 10–14 V vaatien täten paljon enemmän energiaa. FRAM-muistilla datan säilyvyys ja muistin kestävyys ovat huippuluokkaa. Muisti kestää vähintään 100 biljoonaa tallennuskertaa ja se kykenee kymmenen vuoden yhtäjaksoiseen toimintaan. Taulukossa 1 on lisää tilastoja ja vertailua EEPROM- ja flash-muisteihin. [10]

Taulukko 1. Muistiominaisuuksien vertailua

	FRAM	EEPROM	Flash
64 tavun kirjoitusaika muistiin	1,6 μ s	2200 μ s	6400 μ s
64 tavun lukuaika muistiin	1,6 μ s	4,5 μ s	4,5 μ s
Tallennuskerrat	100 * 10 ¹²	500 000	100 000
Kirjoittamisen vaatima jännite	1,5 V	10–14 V	10–14 V
Valmistusaika	-	>3x	3x
Resistanssi gammasäteilylle	Kyllä	Ei	Ei

Texas Instruments on ottanut FRAM-muistiteknologian käyttöön MSP430-tuoteperheessään. MSP430:ssä FRAM-muistia voidaan lukea 8 MHz:n nopeudella ja sen tyypillinen tallennusnopeus voi ylittää nopeuden 2 Mt/s, joka on korkea verrattuna flash-laitteiden noin 14 kt/s tallennusnopeuksiin. Texas Instruments lupaa 10¹⁵ tallennuskertaa mikrokontrollerillensa. Juuri kestävyys ja energiapihiys ovat niitä asioita, miksi FRAM-muisti on valikoitunut MSP430:een, jonka käyttökohteita ovat muun muassa automaatio-sovellukset, kannettavat mittauslaitteet ja lääketieteen sovellukset. [10]

5. PIC24F:N MUISTIRAKENNE

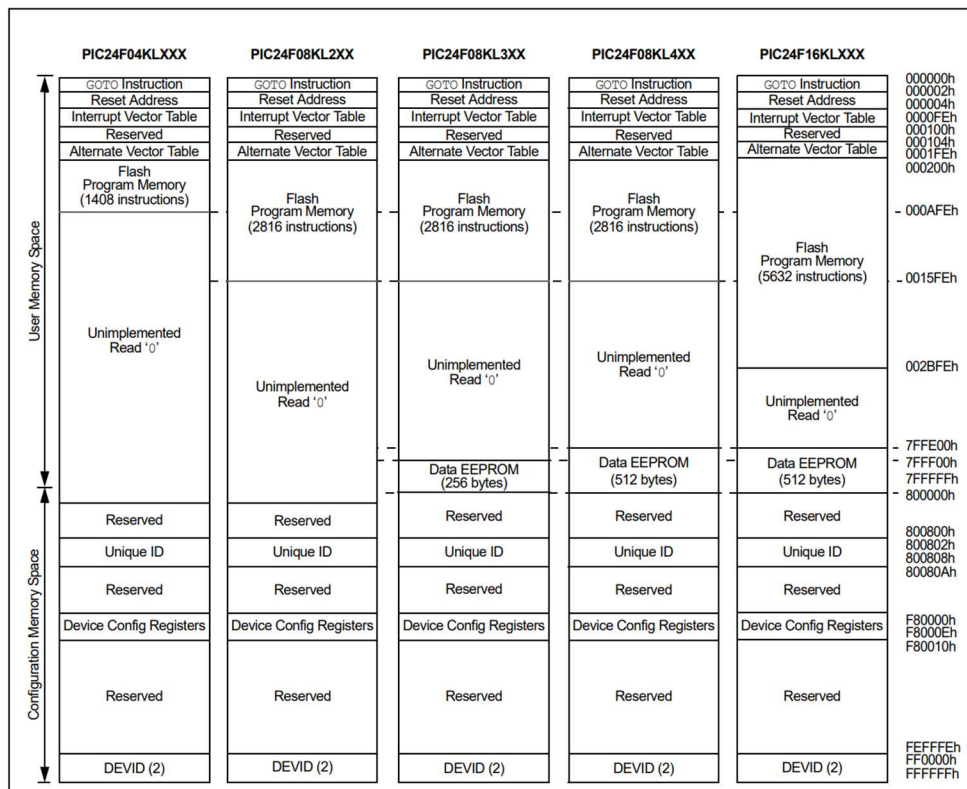
5.1 MIKROKONTROLLERIN YLEISKATSAUS

Microchip Technology Inc. -yhtiön valmistama PIC24F on kustannustehokas 16-bittinen mikrokontrolleri, jonka suurimpia etuja on sen energiatehokkuus. Laitteen kohderyhmä on sovellukset, jotka vaativat matalaa energiankulutusta, mutta enemmän laskentatehoa, kuin vastaavat 8-bittiset mikrokontrollerit.

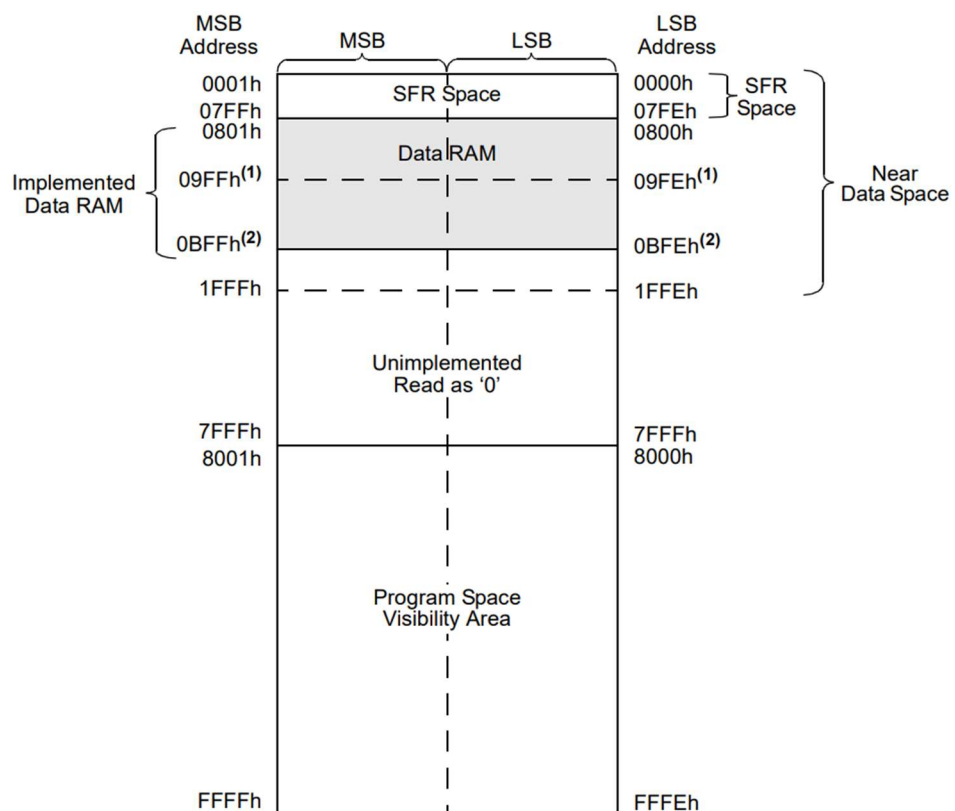
5.2 MUISTIRAKENNE

PIC24F mikrokontrollereille on ominaista erilliset ohjelma- ja datamuistiavaruudet. Arkkitehtuuri antaa myös suoran pääsyn ohjelmamuistiin koodin suorittamisen aikana. Tässä työssä käsitellään erityisesti PIC24F-mikrokontrolleriperheen mallia PIC2416KL402, joka käyttää SRAM-, flash- ja EEPROM-muistityyppejä ohjelma- ja datatietojen tallentamiseen. Tämän mikrokontrollerin ohjelmamuistiavaruus on osoitteellinen 24-bittisellä arvolla, joka on johdettu joko 23-bittisestä ohjelmalaskurista ohjelman suorittamisen aikana tai taulukko-operaatiosta tai datamuistiavaruuden uudelleenmuokkauksen kautta. Käyttäjällä on pääsy ohjelmamuistiavaruuden alempaan puoliskoon, eli alueelle 000000h – 7FFFFFFh. Poikkeuksen tälle tekee TBLRD/TBLWT-operaatiot, jotka mahdollistavat pääsyn asetusbitteihin ja *Device ID* -alueelle. Seuraavalla sivulla on kuvattuna PIC24F16KL402-laitteperheen ohjelmamuistikartta Kuva 7. [11]

Erillinen 16 nbittiä leveä datamuistiavaruus on osoitteellinen yksittäisellä lineaarisella alueella. Datamuistiin pääsee käyttämällä kahta osoitteengenerointiyksikköä (AGU), joista toinen on lukuoperaatiolle ja toinen kirjoitusoperaatiolle. Datamuistikartta on kuvattuna myöhemmin kuvassa 8. [11]



Kuva 7. PIC24F16KL402-laiteperheen ohjelmamuistikartta. Muistialueet eivät ole skaalattuja.



Kuva 8. PIC24F16KL402-laiteperheen datamuistikartta. Muistialueet eivät ole skaalattuja.

Kaikki voimassa olevat osoitteet, *Effective Addressess* (EA), datamuistiavaruudessa osoittavat tavuihin data-alueella antaen osoitealueeksi 64 kilotavua tai 32 tuhatta sanaa. Alempi puolisko datamuistialueesta on implementoitujen muistiosoitteiden käyttämiseen, ja ylempi puolisko on varattu ohjelma-avaruuden näkyvyydelle, *Program Space Visibility* (PSV).

Datamuistissa on kahdeksan kilotavun lähidata-alue, *Near Data Space* (NDS). Tähän muistialueeseen voidaan osoittaa suoraan 13-bittisen osoitekentän kautta ja muualle lähidata-alueella osoittaminen tapahtuu epäsuorasti. Lisäksi koko lähidata-avaruuteen voidaan osoittaa käyttämällä MOV-käskyjä, mikä tukee muistiin osoittamista suoraan 16-bittisen muistikentän kautta, *Memory Direct Addressing* (MDA). Lähidata-alueen alussa olevaa kahden kilotavun erikoistoimintorekisterialuetta, *Special Function Register Space* (SFR), käytettävät PIC24F:n ydin- ja oheismoduulit laitteen toiminnan ohjaamiseksi. [11]

5.3 FLASH-OHJELMAMUISTI

PIC24F16KL402 sisältää flash-ohjelmamuistin tallennuskäyttöön ja sovellusten koodin suorittamiseen. Muisti on luettavissa, kirjoitettavissa ja tyhjennettävissä, kun toimitaan käyttöjännitteen (V_{DD}) ollessa yli 1,8 voltia.

Flash-muistia voi ohjelmoida kolmella eri tavalla. Ensimmäinen tapa on piirin sisäinen sarjaohjelmointi, *In-Circuit Serial Programming* (ICSP). Tällä tavoin ohjelmointi tapahtuu yksinkertaisesti muutamalla koodirivillä, joilla ohjataan ohjelmointikelloa ja ohjelmointidataa ja määritetään muistille teho (V_{DD}) ja maa (V_{SS}). Lisäksi valitaan *Master Clear/Program* -moodi. Tämä antaa asiakkaille mahdollisuuden valmistaa piirilevyjä ohjelmoimattomilla laitteilla, jolloin ohjelmointi voidaan tehdä vasta juuri ennen tuotteen toimittamista, mikä mahdollistaa viimeisimmän tai mukautetun laiteohjelman asentamisen.

Toinen ohjelmointitapa on suoritusaikainen itseohjelmointi, *Run-Time Self Programming* (RTSP), joka toteutetaan käyttämällä *TBLDR* (*Table Read*) ja *TBLWT* (*Table Write*) -käskyjä muistin lukemiseen ja kirjoittamiseen. RTSP:n avulla käyttäjä voi kirjoittaa ohjelmamuistidataa kerralla 32 käskyn, eli 96 tavun lohkoissa. Vastaavasti muistin tyhjentämistoiminnon voi tehdä kerralla 32, 64 ja 128 käskyn (96, 192 ja 384 tavun) lohkoissa. [11]

Kolmas ohjelmointitapa on paranneltu versio ICSP:stä, *Enhanced In-Circuit Serial Programming*, joka käyttää sisäänrakennettua alkulatausohjelmaa ohjelmointiprosessien hallintaan. Parannelun ICSP:n avulla ohjelmamuistin voi tyhjentää, ohjelmoida ja todentaa.

5.4 EEPROM-DATAMUISTI

Haihtumaton EEPROM-datamuisti on PIC24F16KLK402-mikrokontrollerissa erillään ohjelmasta ja haihtuvasta RAM-datamuistista. EEPROM-datamuisti sijoittuu muistikartalla käyttäjän ohjelmamuistiavaruuden yläpäähän, kuten kuva 7 havainnollistaa. Mallissa PIC24F16KLK402 EEPROM-datamuistialueen koko on 256 sanaa. Jokainen sana on suoraan osoitettavissa, ja ne ovat luettavissa sekä kirjoitettavissa. Toisin kuin flash-ohjelmamuistissa, normaalia ohjelman toimintaa ei voi pysäyttää EEPROM-datamuistin ohjelmointi- tai tyhjentämistoiminnan aikana.

EEPROM-datamuistin ohjelmointioperaatioita ohjataan käyttämällä kolmea NVM ohjausrekisteriä: *Nonvolatile Memory Control Register* (NVMCON), *Nonvolatile Memory Key Register* (NVMKEY) ja *Nonvolatile Memory Address Register* (NVMADR). Näistä NVMCON-rekisteri on ensisijainen rekisteri EEPROM-datamuistin ohjelmointi- ja tyhjentämisoperaatioille. NVMKEY-rekisteri on *write-only* -rekisteri, jota käytetään EEPROM-muistipaikkojen tapahtuvien tahtomattomien kirjoittamisten ja tyhjentämisten välttämiseksi. EEPROM-datamuistin voi tyhjentää joko kokonaan tai osittain kolmessa eri koossa: yksi sana, neljä sanaa tai kahdeksan sanaa kerralla. [11]

6. REAALIAIKAKELLOPIIRI JA SEN MUISTI

Reaaliaikakello, *real time clock* (RTC), on kellopiiri, jonka tarkoitus on muistaa kellonaika ja päivämäärä ja laskea aikaa tarkasti. Useimmat RTC-piirit, kuten tässä luvussa esiteltävä DS3231, osaavat laskea karkauspäivät ja vuodet ja pitävät näin päivämäärän oikeana. Tietokone tai mikrokontrolleri kykenee pitämään kellonajan oman laskurinsa avulla, mutta unohtaa ajan heti, kun virta katkaistaan. Reaaliaikakello korjaa tämän puutteen, sillä se on niin vähävirtainen, että se pysyy yhdellä nappiparistolla ajassa vuosia. Reaaliaikakellopiireillä on monesti itse reaaliaikakellon lisäksi muutama kilobitti haihtumatonta muistia. RTC ei niinkään tarvitse muistia toimiakseen, mutta muistiin voidaan tallentaa aikaan perustuva tapahtumarekisteri tai muuta dataa, jonka halutaan olevan haihtumatonta.

Esimerkiksi kuluttaja voi ostaa muutamalla eurolla verkkokaupasta piirin, jossa on integroituna Maxim Integrated Products -yhtiön valmistama DS3231 RTC-moduuli ja Atmelin valmistama haihtumaton AT24C32 EEPROM -muisti, jossa on kahdeksan bittiä leveää muistia 4096 sanan verran. Piirin tarkka ajan laskeminen perustuu sen omaan 32 kHz:n taajuudella värähtelevään kiteeseen. Piiriä voi ohjelmoida, ja siinä on 3 voltin nappiparisto takaamassa ajankulun, vaikka piirille ei muuta virtaa tuotaisikaan. [13][14][15]

Myös itse RTC-moduuliin saatetaan sijoittaa muutama tavu haihtumatonta muistia, kuten Micro Crystal AG on tehnyt RV-3028-C7 -reaaliaikakellomoduulinsa toteutuksessa. Siinä on 51 tavua EEPROM-muistia ja 56 tavua RAM-muistia käyttö- ja asetusmuisteina. [16]

7. POHDINTA

Tämän kirjallisuustutkimuksen tarkoitus oli selvittää eri muistityyppien ominaisuuksia ja niiden eroja.

Muisteja on karkeasti kahdenlaisia: haihtuvia ja haihtumattomia. Haihtuvissa muisteissa on pieni ero luku- ja kirjoitusnopeuksissa ja hinnassa. SRAM-muistin ollessa DRAM-muistia nopeampi, mutta kalliimpi, sitä käytetään paikoissa, joissa tarvitaan nopeaa tiedonsiirtoa.

Haihtumattomissa muisteissa erot ovat isompia. Etenkin ohjelmointiajoissa erot ovat merkittäviä. Lisäksi eroja on tallennuskertojen määrässä, mutta tallennuskertojen määrä ei ole niin suuri rajoittava tekijä, kuin ohjelmointiaika. Myös virrankulutuksessa on pieniä eroja haihtumattomien muistien kesken. Nämä erot johtavat yleensä siihen, että sulautetuissa järjestelmissä on useampaa eri muistia, jotta tietynlainen data tallennetaan muistiin, jonka ominaisuudet ovat tälle datatyypille optimaaliset. Esimerkiksi EEPROM- ja flash-muisteja käytetään parina flashin toimiessa ohjelmamuistina ja EEPROM-muistiin tallennetaan parametreja ja muuta dataa, jota käyttäjä ei jälkeenpäin muokkaa.

FRAM-muisti oli kaikista muistityypeistä selvästi mielenkiintoisin, sillä siihen kohdistuu odotuksia sulautetuissa järjestelmissä käytettävien muistitekniikoiden mullistamisesta. FRAM:in ollessa ominaisuuksiltaan ylivertainen muihin muisteihin verrattuna, se saattaa hyvinkin korvata muut muistityypit hoitaen yksinään erilaisten datatyyppien käsittelyn.

Tämän tutkimuksen yhtenä suurimmista haasteista oli löytää tuoretta tietoa, sillä useimmat käsittelyssä olleet muistityypit ovat olleet jo vuosikymmeniä tutkimuksen alla. Lisäksi uudemmissa artikkeleissa kyseisien muistien tuntemusta pidettiin enimmäkseen itsestäänselvyytenä. Täten hyviä lähteitä muistien ominaisuuksien esittämiseen oli hankala löytää. Tähän FRAM oli virkistävä poikkeus. Toisaalta yksittäisiä tiedonjyväsiä oli helppo varmentaa käyttämällä avoimia tietosanakirjoja ja tieteellisiä hakukoneita. Muisteista on tehty tutkimusta niin paljon, että jos jokin asia esiintyy useassa artikkelissa, se myös todennäköisesti pitää paikkansa. Tästä esimerkkinä käyttökohteet, joihin eri muistityyppejä käytetään.

8. YHTEENVETO

Sulautetut suorittimet käsittelevät paljon erilaista dataa, jota tallennetaan kyseessä olevalle datatyypille sopivaan muistiin. Haihtumattomista muisteista SRAM:ia käytetään suuren luku- ja kirjoitusnopeutensa vuoksi välimuistina, jota on sekä itse suorittimen sisällä, että sen välittömässä läheisyydessä. DRAM on toinen haihtuvista muistityypeistä, mutta sitä ei juurikaan käytetä sulautetuissa järjestelmissä.

Alkulatauskoodi, ohjelmakoodi, muuttujat, kalibrointiarvot ovat datatyyppejä, jotka tallennetaan haihtumattomaan muistiin. Tällaisia haihtumattomia muisteja ovat esimerkiksi flash-, EEPROM- ja FRAM-muistit. EEPROM-muistiin tallennetaan dataa, jota ei tule enää muokata. Tällaista dataa ovat parametrit, alkulatauskoodi ja ohjelmakoodi, jota ei päivitetä. Flash- ja FRAM-muisteihin voidaan tallentaa myös päivitettävää dataa, sillä näitä muisteja voi uudelleenohjelmoida.

Työssä tutkittiin Microchip Technologyn PIC24F-mikrokontrollerin muistiavaruuden sisältöä. Tämä mikrokontrolleri käyttää SRAM-, flash- ja EEPROM-muistityyppejä ohjelma- ja datatietojen tallentamiseen, ja sillä on erilliset ohjelma- ja datamuistiavaruudet. Flash-ohjelmamuistiavaruus on osoitteellinen 24-bittisellä arvolla ja EEPROM-datamuistiavaruus on 16 bittiä leveä.

Lopuksi työssä käsiteltiin reaaliaikakellopiirejä, joiden tarkoitus on muistaa kellonaika ja päivämäärä ja laskea aikaa tarkasti. Reaaliaikakellopiiri ei niinkään tarvitse muistia toimiakseen, mutta niissä on yleensä jonkin verran haihtumatonta muistia, johon voidaan tallentaa dataa, jonka halutaan olevan haihtumatonta.

9. LÄHTEET

- [1] Melear C. (1994) Integrated memory elements on microcontroller devices. Proc WESCON - Idea/Microelectron , Conf Rec. 1994:507-14.
- [2] Chang IJ, Kang Y, Kim Y. (2019) Channel length biasing for improving read margin of the 8T SRAM at near threshold operation. Electronics (Switzerland). 2019;8(6).
- [3] Jacob B, Ng SW, Wang DT. (2008) Memory systems : cache, DRAM, disk. Burlington, MA: Morgan Kaufmann Publishers.
- [4] Conrad E, Misenar S, Feldman J. (2014) Eleventh Hour CISSP: Study Guide, Second Edition. second ed.
- [5] R. Bez, E. Camerlenghi, A. Modelli, A. Visconti. (2009) Introduction to flash memory. Proceedings of the IEEE. 2003;91(4):489-502.
- [6] Wong W. (2002) Flash-based microcontrollers are rapidly taking charge. Electron Des. 2002;50(25):53-6.
- [7] Jew T. (2015) Embedded Microcontroller Memories: Application Memory Usage. IEEE Int Mem Workshop, IMW.
- [8] Basic Serial EEPROM Operation (1993). Microchip Technology Inc. URL: <http://ecee.colorado.edu/~mcclurel/man536.pdf>
- [9] FRAM – New Generation of Non-Volatile Memory (2009). Texas Instruments Incorporated. URL: <https://www.ti.com/lit/ml/szzt014a/szzt014a.pdf>
- [10] Priya T, William Goh. (2012) MSP430™ FRAM Quality and Reliability.
- [11] PIC24F16KL402 FAMILY. Low-Power, Low-Cost, General Purpose 16-Bit Flash Microcontrollers with XLP Technology. Microchip Technology Inc. (2019) URL: <http://ww1.microchip.com/downloads/en/DeviceDoc/PIC24F16KL402-Family-Data-Sheet-DS30001037D.pdf>
- [12] Kinam Kim, Yoon J. Song. (2003) Integration technology for ferroelectric memory devices.
- [13] 2-Wire Serial EEPROM. AT24C32/64 (2003). Atmel Corporation. URL: <http://ww1.microchip.com/downloads/en/devicedoc/doc0336.pdf>

- [14] DS3231. Extremely Accurate I2C-Integrated RTC/TCXO/Crystal. Maxim Integrated Products, Inc. (2005). URL:
<https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>

- [15] RV-3028-C7. Extreme Low Power RTC Module (2019). Micro Crystal AG. URL:
<https://www.microcrystal.com/fileadmin/Media/Products/RTC/Datasheet/RV-3028-C7.pdf>